Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Previously Presented) A processor, comprising:

a plurality of pipelined functional units for executing instructions;

a centralized scheduler, coupled to the plurality of functional units, wherein the centralized scheduler is programmed to receive via an instruction buffer and an instruction decoder at least two separate instruction groups, in a first stage map each of the at least two separate instruction groups to at least a portion of the functional units independently of each other in which the centralized scheduler treats each instruction group as having full access and availability to the plurality of pipelined functional units, and based at least in part on functional unit availability and instruction dependencies, perform a merging and remapping of the at least two separate instruction groups to the at least a portion of the functional units in a second stage to ensure that no resource conflict occurs between the plurality of pipelined functional units.

2. (Previously Presented) The processor of claim 1, wherein the centralized scheduler is programmed to deliver the instructions to the portion of functional units following merging and remapping.

3. (Canceled)

4. (Previously Presented) The processor of claim 1, wherein the at least a portion of the functional units execute instructions from the at least two instruction groups.

5. (Original) The processor of claim 1, wherein the instruction groups follow a simultaneous multi-threading structure.

6. (Original) The processor of claim 1, wherein the instruction groups are prioritized to prevent pipeline failures during execution of instructions.

7. (Previously Presented) A machine-readable medium having stored thereon a plurality of executable instructions, the plurality of instructions comprising instructions to:

receive at least two separate instruction groups in a scheduler via an instruction buffer and an instruction decoder;

in a first stage of the scheduler, map each of the at least two separate instruction groups to at least a portion of functional units independently of each other; and

based at least in part on functional unit availability and instruction dependencies, perform a merging and remapping of the at least two separate instruction groups to the at least a portion of functional units in a second stage of the scheduler.

8. (Original) The medium of claim 7, wherein said instructions include instructions to deliver the instructions to the portion of functional units following merging and remapping.

9. (Canceled)

10. (Previously Presented) The medium of claim 7, wherein the at least a portion of functional units execute instructions from the at least two instruction groups.

11. (Original) The medium of claim 7, wherein the instruction groups follow a simultaneous multi-threading structure.

12. (Original) The medium of claim 7, wherein the instruction groups are prioritized to prevent pipeline failures during execution of instructions.

13. (Previously Presented) A method for dispersing instructions to executed by a processor, comprising:

receiving at least two separate instruction groups in a scheduler via an instruction buffer and an instruction decoder;

in a first stage of the scheduler, mapping each of the at least two separate instruction groups to at least a portion of functional units independently of each other; and

based at least in part on functional unit availability and instruction dependencies, performing a merging and remapping of the at least two separate instruction groups to the at least a portion of functional units in a second stage of the scheduler.

14. (Original) The method of claim 13, further comprising: delivering the instructions to the portion of functional units following merging and remapping.

15. (Previously Presented) The method of claim 13, wherein the at least a portion of functional units execute instructions from the at least two instruction groups.

16. (Canceled)

17. (Original) The method of claim 13, wherein the instruction groups follow a simultaneous multi-threading structure.

18. (Original) The medium of claim 13, wherein the instruction groups are prioritized to prevent pipeline failures during execution of instructions.